Dilnoza Saidova

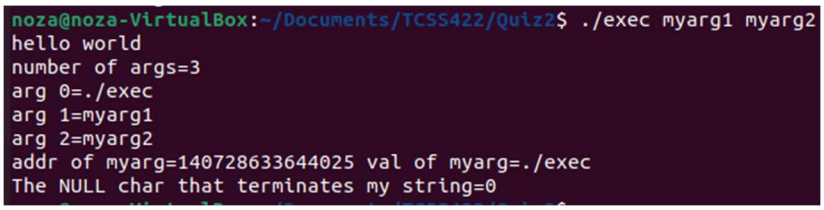
TCSS422: Operating Systems

January 17, 2023

Quiz 2

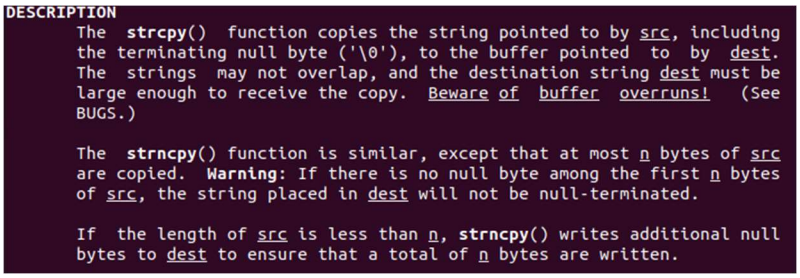
Question 1. What do you see? What you just did is not possible in Java. Why not?
Note: Please provide the screenshot of your output (whole terminal containing your output) and answer these two questions.

The line printed the NULL ASCII character that terminates the string, which is `0`. This is not possible in Java because this would cause a NullPointerException – a situation where uninitialized object is attempted to be accessed/modified.



Question 2. What is strcpy()? Hint: check out the man page using command 'man strcpy'.

`strcpy()` is used to copy the content of one string to another. The `strcpy()` function doesn't create a string (or character array), it only copies a string (or character array).



Question 3. Does creating a string array on the heap and assigning each of the strings in the array to point to existing strings give us our own unique copy of the strings? Note: Please provide the screenshot of your output and answer the question.

Creating a string on the heap and assigning each of the strings in the array to point to existing strings doesn't give us our own unique copy of the strings as it points to the elements in the array it's pointing to (same memory address). In other words, pointers are created.

```
noza@noza-VirtualBox:~/Documents/TCSS422/Quiz2$ ./exec myarg1 myarg2
hello world
number of args=3
arg 0=./exec
arg 1=myarg1
arg 2=myarg2
addr of myarg=140730080150521 val of myarg=./exec
The NULL char that terminates my string=0
myarg char 0=46 .
myarg char 1=47 /
myarg char 2=101 e
myarg char 3=120 x
myarg char 4=101 e
myarg char 5=99 c
myarg char 6=0
ourarray arg 0=myarg2
ourarray arg 1=myarg1
ourarray arg 2=./exec
ourarray[0]=mbarg2 -- argv[0]=./exec
ourarray[1]=myarg1 -- argv[1]=myarg1
ourarray[2]=./exec -- argv[2]=mbarg2
```

Question 4. When we change one character of ourarray[0] to 'b', do we see this change in argv as well? Yes or no? Note: Please provide the screenshot of your output (whole terminal containing your output) and answer the question.

No.

```
noza@noza-VirtualBox:~/Documents/TCSS422/Quiz2$ ./exec myarg1 myarg2
hello world
number of args=3
arg 0=./exec
arg 1=myarg1
arg 2=myarg2
addr of myarg=140736191296505 val of myarg=./exec
The NULL char that terminates my string=0
myarg char 0=46 .
myarg char 1=47 /
myarg char 2=101 e
myarg char 3=120 x
myarg char 4=101 e
myarg char 5=99 c
myarg char 6=0
ourarray arg 0=
ourarray arg 1=
ourarray arg 2=
ourarray[0]= -- argv[0]=./exec
ourarray[1]= -- argv[1]=myarg1
ourarray[2]= -- argv[2]=myarg2
```

Question 5. When searching the internet, what methods do you see described to read strings from the console (user)?

`scanf`, `fgets`, and `fscanf`.

Question 6. Question 6. What is the ASCII integer value for the last character ALWAYS? Note: Please provide the screenshot of your output (whole terminal containing your output) and answer the question.

The ASCII integer value for the last character is always 0 (zero) regardless of the input.

```
noza@noza-VirtualBox:~/Documents/TCSS422/Quiz2$ ./execvp
cmd->test
char 4=0
noza@noza-VirtualBox:~/Documents/TCSS422/Quiz2$ ./execvp
cmd->dilnoza
char 7=0
noza@noza-VirtualBox:~/Documents/TCSS422/Quiz2$ ./execvp
cmd->alwayszero
char 10=0
noza@noza-VirtualBox:~/Documents/TCSS422/Quiz2$ ./execvp
cmd->a
char 1=0
noza@noza-VirtualBox:~/Documents/TCSS422/Quiz2$
```

Question 7. How can we assign the second pointer in our character array?

The second pointer in the array can be added into the array by adding 1 to `args`: `*(args + 1) = arg1;`

```
noza@noza-VirtualBox:~/Documents/TCSS422/Quiz2$ grep "*(" execvp.c
  *(args) = cmd;
  *(args + 1) = arg1;
  *(args + 2) = arg2;
  *(args + 3) = file;
  *(args + 4) = '\0';
    printf("i=%d args[i]=%s\n",i,*(args + i));
```

Question 8. Using our string array, how can we reference just the string of the command (e.g., grep, wc, ls themselves)?

Using the string array, the string of the command can be referenced by using the grep as it prints the lines that match the specified patterns.

Question 9. Please provide the screenshot of your output. Is the output as what you expected? Why we have this output using what we have discussed in the lecture of process API? [2 points for screenshot and 2 for explanations].

The given output is what I expected. I put in the input strings first and then the `char 4=0` displays the number of chars in the first argument (`cmd->grep`). The lines before ending `0` show what indexes `i` is on and displaying the strings at those indexed. The ending `0` shows the program's finished executing.

```
noza@noza-VirtualBox:~/Documents/TCSS422/Quiz2$ man grep
noza@noza-VirtualBox:~/Documents/TCSS422/Quiz2$ vim execvp.c
noza@noza-VirtualBox:~/Documents/TCSS422/Quiz2$ make
gcc  -pthread -I. -Wall     execvp.c   -o execvp
noza@noza-VirtualBox:~/Documents/TCSS422/Quiz2$ ./execvp
cmd->grep
arg1->-c
arg2->the
file->execvp.c
char 4=0
i=0 args[i]=grep
i=1 args[i]=-c
i=2 args[i]=the
i=3 args[i]=execvp.c
i=4 args[i]=(null)
0
```